

CLAIMS

What is claimed is:

1. A method of loading atomized computer program code and data on demand, comprising:
 - 5 loading, from an atom database, into a memory, an atom defining code or data in a fine-grained, individually addressable manner, the atom comprising:
 - an atom identifier;
 - computer program code or data information;
 - 10 computer program code or data reference information; and
 - modifying the computer program code or data information by transforming a referencing atom identifier into a memory address.
2. The method of Claim 1 wherein the memory address is a memory address of a stub routine that loads the referenced atom and jumps to the referenced atom.
 - 15
3. The method of Claim 2 wherein the memory address to the stub routine is overwritten by an updated memory address directly referencing the referenced atom.
 - 20
4. The method of Claim 2 wherein the memory address to the stub routine is overwritten for all loaded atoms referencing the referenced atom.
 - 25
5. The method of Claim 1 wherein the step of modifying further comprises:
 - encoding the memory address such that a referenced data atom is not loaded until actually accessed at runtime.

6. The method of Claim 5 wherein the step of encoding comprises:
multiplying the referencing atom identifier by two and incrementing
the referencing atom identifier by one to produce a lazy data atom identifier.
7. The method of Claim 1 further comprising the step of:
reordering the atoms in a disk-based atom database for efficient
access based upon analyzing a runtime pattern of previous atom loading
experiences.
8. The method of Claim 1 further comprising the step of:
updating at least one of the atoms in the atom database by replacing
the computer program code or data information and the computer program
code or data reference information.
9. The method of Claim 1 further comprising the step of:
adding a new atom to the atoms in the atom database.
10. The method of Claim 1 further comprising the step of:
deleting a selected atom from the atoms in the atom database.
11. The method of Claim 1 wherein the loading step further comprises:
decompressing the computer program code or data information stored
in a compressed format.
12. The method of Claim 1 wherein loaded atoms are removed from memory
based upon a predetermined usage threshold.
- 20 13. The method of Claim 1 wherein execution of a loaded atom begins prior to
loading all atoms referenced by the loaded atom.
14. The method of Claim 1 further comprising the step of:

modifying the computer program code or data reference information to effect an eager or lazy loading technique.

15. The method of Claim 14 wherein the step of modifying determines the loading technique based upon analyzing a runtime pattern of previous atom loading experiences.
5
16. The method of Claim 1 wherein a loaded atom is shared between a plurality of executable processes by way of a read-only buffer.
17. A method for atomizing computer program code and data, comprising:
receiving the computer program code and data in an object code
10 format defining individually addressable code and data;
extracting computer program code and data information from the computer program code and data in an object code format;
extracting computer program code and data reference information from the computer program code and data in an object code format;
15 modifying the computer program code and data reference information to use atom identifiers; and
storing the computer program code and data information and computer program code and data reference information in an atom comprising:
20 an atom identifier;
computer program code or data information; and
computer program code or data reference information.
18. An apparatus loading atomized computer program code and data on demand, comprising:
25 an atom comprising:
an atom identifier;
computer program code or data information;
computer program code or data reference information; and

5 an atom management program unit loading, from an atom database, into a memory, the atom defining code or data in a fine-grained, individually addressable manner, and modifying the computer program code or data information by transforming a referencing atom identifier into a memory address.

19. An apparatus atomizing computer program code and data, comprising:
a receiver receiving the computer program code and data in an object code format defining individually addressable code and data;
an extractor extracting computer program code and data information from the computer program code and data in an object code format, and extracting computer program code and data reference information from the computer program code and data in an object code format;
- 10 a modifier modifying the computer program code and data reference information to use atom identifiers; and
- 15 a storage unit storing the computer program code and data information and computer program code and data reference information in an atom comprising:
an atom identifier;
computer program code or data information; and
- 20 computer program code or data reference information.
- 25 20. An apparatus loading atomized computer program code and data on demand, comprising:
a means for loading, from an atom database, into a memory, an atom defining code or data in a fine-grained, individually addressable manner, the atom comprising:
an atom identifier;
computer program code or data information;
computer program code or data reference information; and

a means for modifying the computer program code or data information by transforming a referencing atom identifier into a memory address.

21. An apparatus atomizing computer program code and data, comprising:
 - 5 a means for receiving the computer program code and data in an object code format defining individually addressable code and data;
 - a means for extracting computer program code and data information from the computer program code and data in an object code format;
 - a means for extracting computer program code and data reference information from the computer program code and data in an object code format;
 - 10 a means for modifying the computer program code and data reference information to use atom identifiers; and
 - a means for storing the computer program code and data information and computer program code and data reference information in an atom comprising:
 - 15 an atom identifier;
 - computer program code or data information; and
 - computer program code or data reference information.
- 20 22. A computer program product comprising:
 - a computer usable medium for loading atomized computer program code and data on demand;
 - a set of computer program instructions embodied on the computer usable medium, including instructions to:
 - 25 load, from an atom database, into a memory, an atom defining code or data in a fine-grained, individually addressable manner, the atom comprising:
 - an atom identifier;
 - computer program code or data information;
 - computer program code or data reference information; and

modify the computer program code or data information by transforming a referencing atom identifier into a memory address.

23. A computer program product comprising:
 - a computer usable medium for atomizing computer program code and data;
 - 5 a set of computer program instructions embodied on the computer usable medium, including instructions to:
 - receive the computer program code and data in an object code format defining individually addressable code and data;
 - extract computer program code and data information from the computer program code and data in an object code format;
 - 10 extract computer program code and data reference information from the computer program code and data in an object code format;
 - modify the computer program code and data reference information to use atom identifiers; and
 - 15 store the computer program code and data information and computer program code and data reference information in an atom comprising:
 - an atom identifier;
 - computer program code or data information; and
 - computer program code or data reference information.
- 20 24. A computer data signal embodied in a carrier wave comprising a code segment for loading atomized computer program code and data on demand, including instructions to:
 - load, from an atom database, into a memory, an atom defining code or data in a fine-grained, individually addressable manner, the atom comprising:
 - an atom identifier;
 - computer program code or data information;
 - computer program code or data reference information; and
 - 25 modify the computer program code or data information by transforming a referencing atom identifier into a memory address.
- 30

25. A computer data signal embodied in a carrier wave comprising a code segment for atomizing computer program code and data, including instructions to:
- receive the computer program code and data in an object code format defining individually addressable code and data;
- 5 extract computer program code and data information from the computer program code and data in an object code format;
- extract computer program code and data reference information from the computer program code and data in an object code format;
- 10 modify the computer program code and data reference information to use atom identifiers; and
- store the computer program code and data information and computer program code and data reference information in an atom comprising:
- an atom identifier;
- 15 computer program code or data information; and
- computer program code or data reference information.
26. A memory for storing data for access by a computer program being executed on a data processing system, comprising:
- a data structure stored in the memory representing an atom database,
- 20 the data structure comprising:
- a header; and
- an atom map, the atom map comprising:
- an atom map header; and
- an atom map body, the atom map body comprising:
- 25 an atom map array referencing an atom offset sequence, the atom offset sequence comprising:
- a first atom file offset;
- a delta coded file offset array; and referencing
- an atom comprising:
- 30 atom flags;
- atom number information;

encoded atom references; and
encoded raw atom bytes.

27. The memory of Claim 26 wherein the atom map provides a mapping of the
5 atom identifiers to symbolic names.